

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

NIGHTINGALE

Atty. Ref.: 550-510

Serial No. 10/764,495

Group: 2128

Filed: January 27, 2004

Examiner: Alhija, Saif A.

For: APPARATUS AND METHOD FOR PERFORMING HARDWARE
AND SOFTWARE CO-VERIFICATION TESTING

Before the Board of Patent Appeals and Interferences

REPLY BRIEF FOR APPELLANT

**On Appeal From Final Rejection
From Group Art Unit 2128**

John R. Lastova
NIXON & VANDERHYE P.C.
11th Floor, 901 North Glebe Road
Arlington, Virginia 22203-1808
(703) 816-4025
Attorney for Appellants
Nightingale and
ARM Limited



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of

NIGHTINGALE

Serial No. 10/764,495

Filed: January 27, 2004

Atty. Ref.: 550-510

TC/A.U.: 2128

Examiner: Alhija, Saif A.

For: APPARATUS AND METHOD FOR PERFORMING HARDWARE
AND SOFTWARE CO-VERIFICATION TESTING

November 9, 2007

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

Sir:

This is a reply to the Examiner's Answer dated September 11, 2007.

THE REJECTION UNDER 35 U.S.C. §101

The Answer does not address the points made in the Brief. Instead, the Examiner simply argues that in order for claimed subject matter that can be implemented by a computer to be statutory, the specification must state that "the processing unit **can only be implemented in hardware**" (emphasis in the original on page 19 of the Answer). The Examiner cites no authority for this alleged statutory subject matter test. Nor is there any.

The Board's attention is also directed to page 21, lines 3-21 of the specification that describe:

a general purpose computer 500 of the type that may be used to implement the above described techniques. The general purpose computer 500 includes a central processing unit 502, a random access memory 504, a read only memory 506, a network interface card 508, a hard disk drive 510, a display driver 512 and monitor 514, and a user input/output circuit 516 with a keyboard 518 and mouse 520 all connected via a common bus 522. In operation, the central processing unit 502 will execute computer program instructions that may be stored in one or more of the random access memory 504, the read only memory 506 and the hard disk drive 510 or dynamically downloaded via the network interface card 508...The computer program may be stored and distributed on a recording medium or dynamically downloaded to the general purpose computer 500. When operating under control of an appropriate computer program, the general purpose computer 500 can perform the above described techniques and can be considered to form an apparatus for performing the above described techniques. The architecture of the general purpose computer 200 can vary considerably, and Figure 6 is only one example.

Clearly, a computer, which is a tangible and useful hardware device, may be configured using a software program to implement features recited in the claims.

Moreover, the invention yields a useful, concrete, and tangible result--the co-verification of hardware and software on a system under verification as explained. That one or more of the features set out in the claim could be embodied as software used to control a computer does not affect that position. There is still a useful, concrete, and tangible result produced. Effective and efficient testing and validation of data processing system designs is an important factor in the design process that suffers from a number of technical problems, as described in the introduction of the present application. The claimed technology enables co-verification of hardware and software to be performed in

a coordinated manner, providing significant benefits over existing techniques. The claims recite statutory subject matter.

THE PRIOR ART REJECTION BASED ON RAJSUMAN

Page 23 of the Examiner's Answer refers to column 7, lines 15 to 25 in Rajsuman which identifies a basic structure in the design validation station 50 that may be used for software/hardware co-development/verification. The Examiner also makes reference to various sections of Rajsuman which describe standard debugging of faults in the cores of the SoC. But such standard debugging activity in the context of software/hardware co-development/verification fails to teach the features of the independent claims. (The distinctions made with respect to claim 1 below apply to all the independent claims).

The detailed interrelationship of features in those claims enables coordinated execution of the software routines with the sequence of verification tests, as recited in the last two lines of claim 1, for example. To assist in understanding the interrelationship of the features in claim 1, a copy of Figure 3 is attached with the Reply Brief marked up to show correspondence between the language in claim 1 and the non-limiting example features shown in Figure 3.

Significantly, the debugger is not being used in the standard manner to debug faults on the CPU. the CPU is a known component and assumed to be executing correctly (i.e., has already been debugged). Rather, the debugger in combination with the processing unit synchronize hardware and software activities to enable a sequence of verification tests to be performed to test correct operation of the software and hardware of the system in combination.

The Examiner argues that this synchronization feature is not recited in the claims (see page 21, second paragraph). The Examiner also refers to a synchronized clock unit in Rajsuman (element 75 in Figure 5. But that clock unit 75 is not what is claimed.

Although the word “synchronization” is not used in claim 1, the interrelationship of features which gives rise to the above-described synchronization is explicitly recited in the claims. In addition to a plurality of signal interface controllers coupled to the system under verification, claim 1 recites a debugger signal interface controller which interfaces with a debugger. A test manager, coupled to both the signal interface controllers and the debugger signal interface controller, transfers test controlling messages (over path 30 as shown in Figure 3) to both the signal interface controllers and the debugger signal interface controller identifying test actions to be performed. With regard to any test actions received by the debugger signal interface controller, claim 1 explains that these involve transferring at least one of one or more stimulus signals and one or more response signals between the debugger and the debugger signal interface controller during performance of the sequence of verification tests. Through use of the debugger signal interface controller, the test manager can control the debugger. Based on these control signals received via the debugger signal interface controller, the debugger then controls operation of a processing unit associated with the system under verification, that processing unit executing software routines.

Accordingly, this control path (from test manager 20 through path 30 to debugger signal interface controller 300, path 305, debugger 310 and path 315 to the CPU 150 as shown in the non-limiting example of Figure 3) allows the test manager to control the

operation of the processing unit, as stated in the last paragraph of claim 1, in order to coordinate the execution of the software routines with the sequence of verification tests. Such control over the execution of the software routines executing on the CPU has previously not been possible and provides a significant improvement in the level of control when seeking to perform hardware and software co-verification on a system under verification.

The Examiner points to general comments in Rajsuman that imply some sort of standard debugging functionality. But Rajsuman does not disclose using that debugging functionality in the manner described in claim 1. The claimed debugger is used with a debugger signal interface controller to provide the test manager with a mechanism for gaining control over the software executing on the CPU, and in this way, provides the test manager with a mechanism to synchronize hardware and software activities. The claimed debugger can perform standard debugging, but it also is used in combination with an associated debugger signal interface controller to perform an entirely different task.

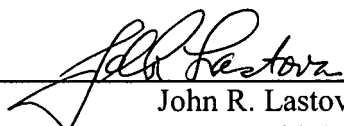
The Answer also fails to demonstrate where Rajsuman discloses a debugger signal interface controller for allowing the test manager to control a debugger in the manner described in claim 1.

For the reasons explained above and in the Brief, the Board should reverse the final rejection.

NIGHTINGALE
Serial No. 10/764,495
November 9, 2007

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 
John R. Lastova
Reg. No. 33,149

JRL:maa
901 North Glebe Road, 11th Floor
Arlington, VA 22203
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

① ≡ signal interface controllers

② ≡ debugger

③ ≡ processing unit

④ ≡ debugger signal interface controller

⑤ ≡ test manager

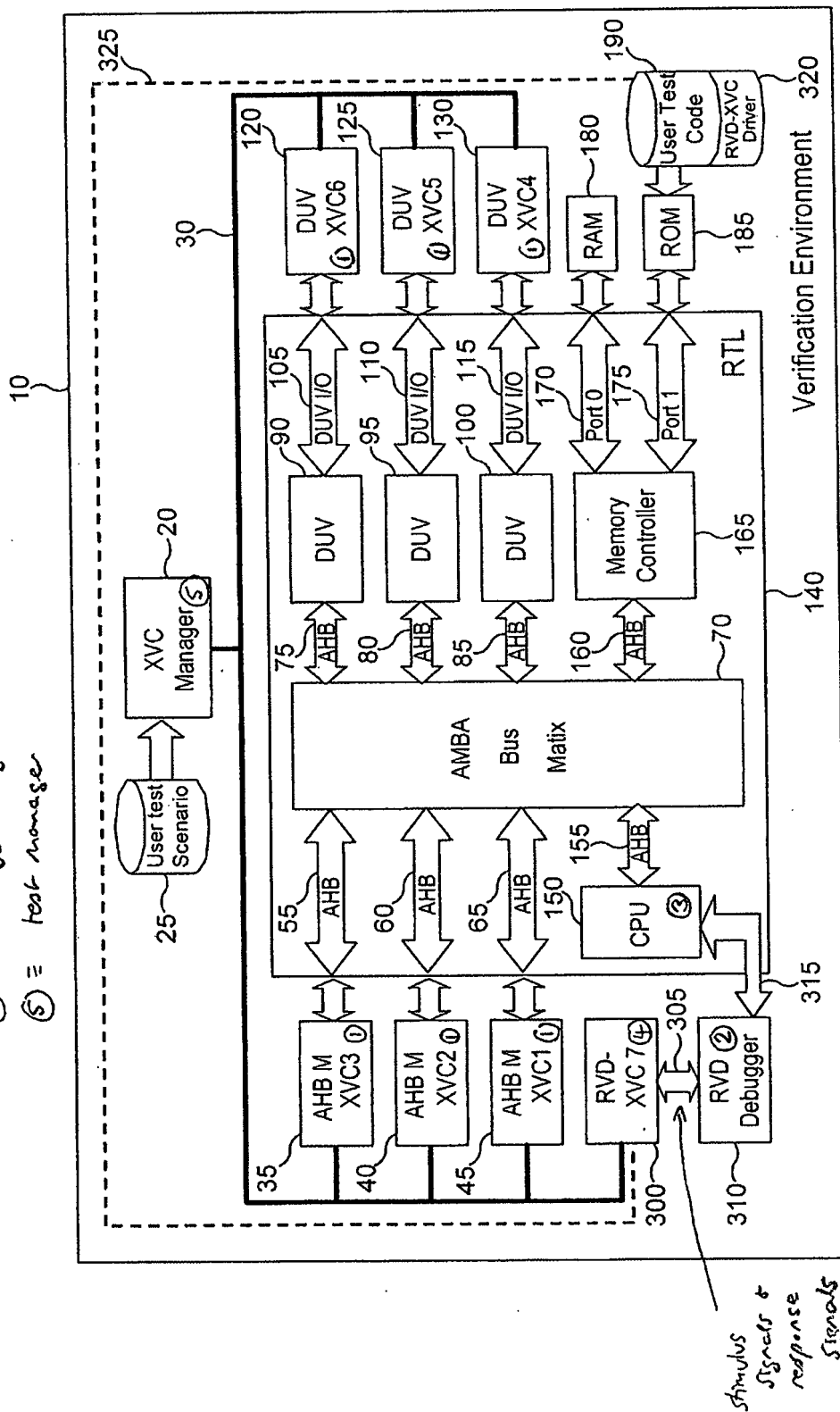


FIG. 3